doi:10.19306/j.cnki.2095-8110.2020.01.004

受限资源下制导武器末制导机器视觉技术研究

赵晓冬[1],车　军[2],张洵颖[1],程雪梅[3]

(1. 西北工业大学无人系统技术研究院,西安 710072;
2. 航空工业西安飞行自动控制研究所,西安 710076;
3. 西北工业大学 365 研究所,西安 710072)

摘　要:针对精确制导武器末制导机器视觉技术应用需求,研究了基于卷积神经网络的、针对复杂背景及小目标的自主目标检测识别算法,并分别进行了网络性能评估和硬件资源需求定量评估。针对最优算法,提出了基于嵌入式受限资源下的高精度神经网络压缩算法,并对算法进行了普适性评估。基于 GPU 嵌入式平台,实现 TensorRT 路线网络优化,并在速度和精度两方面均衡考虑下,对裁剪与量化算法进行了详细实验验证。实验结果表明,高精度神经网络压缩算法在硬件资源受限条件下,可以有效提升推理速度,最终经算法优化后的网络结构,可以获得 3 倍以上的速度提升,网络精度损失小于 5%。

关键词:自主检测识别;神经网络性能评估;硬件资源定量评估;网络高精度压缩;嵌入式应用

中图分类号:V448　　**文献标志码**:A　　**开放科学(资源服务)标识码(OSID)**:

文章编号:2095-8110(2020)01-0026-08

Research on Terminal Guidance Machine Vision Technology for Guided Weapons with Limited Hardware Resources

ZHAO Xiao-dong[1], CHE Jun[2], ZHANG Xun-ying[1], CHENG Xue-mei[3]

(1. Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China;
2. AVIC Xi'an Flight Automatic Control Research Institute, Xi'an 710076, China;
3. Institute NO. 365, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Aiming at the application requirement of machine vision technology in the terminal guidance of precision guided weapon, the autonomous target detection and recognition algorithms based on the convolutional neural network for complex background and small target are studied, and the network performance evaluation and hardware resource requirement quantitative evaluation are carried out respectively. Aiming at the optimal algorithm, high-precision neural network compression algorithm based on embedded limited resources is proposed, and the general applicability of the algorithm is evaluated. Based on the GPU embedded platform, the TensorRT network optimization is realized, Furthermore, the balance of speed and precision is verified by detailed experiments. The experimental results show that the high-precision neural network compression algorithm can effectively improve the inference speed under the condition of limited hardware resources. Finally, the network structure optimized by the algorithm can achieve more than three

收稿日期:2019-08-30;**修订日期**:2019-09-23

作者简介:赵晓冬(1986-),女,博士,助理研究员,主要从事人工智能算法及嵌入式机器视觉方面的研究。
E-mail:xdzhao@nwpu.edu.cn

times the speed improvement, and the network accuracy loss is less than 5%.

Key words: Autonomous detection and recognition; Neural network performance evaluation; Quantitative evaluation of hardware resources; High-precision network compression; Embedded applications

0　引言

在现代战争当中,精确制导武器的成功研制,促使定点攻击作战技术迈上新的台阶。光学制导技术作为精确制导的重要组成部分,是决定其作战性能的重要因素。光学制导包括可见光电视、红外、激光、光纤及复合制导等,其中可见光电视、红外和复合制导都属于图像处理与机器视觉的范畴。可见光电视制导由弹上成像系统负责完成目标的探测与识别,较为成熟的包括“GBU-15”制导炸弹、“KAB-1500KR”以及“AGM-144”反坦克弹等制导武器。红外制导利用热成像探测原理,实现目标的检测与识别,较为成熟的包括“萨姆-7”、“红缨-5”和美国“战斧”巡航导弹 BlockIV 等。复合制导采用多模式复合方式,取长补短,具有代表性的有美国的 RAM 航空弹和 AARGM 导弹,分别采用雷达与红外复合、微波与红外复合的制导方式。

在精确制导武器末制导过程中,国内军事上人工智能算法依旧处于无法落地的阶段。美国洛克希德·马丁公司的远程反舰导弹 LRASM,已经成功完成了多次靶试任务。LRASM 基于一款较为成熟的空间导弹进行研制,旨在依靠自身人工智能处理器,在舰队中检测并摧毁特定军事目标。该型导弹于 2017 年 12 月成功击中海上移动目标,标志着其技术已完全成熟,达到列装标准。同年 2 月,俄罗斯武器制造商与国防官员宣布开发内置人工智能的新一代武器,该类智能武器可自主选择目标。2019 年 6 月,以色列拉斐尔公司已成功将人工智能集成到 Spice 炸弹中,在实现目标自主识别的基础上,加入了人工智能及场景匹配技术。凭借人工智能和深度学习技术,该武器可以识别移动的地面目标,并成功将其与其他物体及地形进行区分。

针对末制导视觉处理方面的经典算法研究成果较多,包括差分图像法[1]、光流场算法[2]、统计模型算法[3]、小波变换算法[4]等。差分法是指利用多帧图像计算出差分图像,并将其对应像素进行相乘,用以消除伪运动图像信息;光流场算法利用基于特征信息的光流场进行运动目标检测,同时利用图像分割获得目标的完整轮廓形状;统计模型首先对运动场进行粗略估计,并根据马尔可夫场理论,构造间断点,实现目标检测;小波变换利用在多尺度上计算由方向、尺度等参数构成的向量来实现目标检测。其中,差分图像法与光流法在工程实现当中应用较多。在信息化作战方面,智能化电子战的概念不断涌现。利用人工智能感知技术获取战场信息,并将信息应用到精确制导武器末制导阶段当中,是一种全新概念的作战方式。从算法理论到工程应用的鸿沟,直接影响武器装备智能化的升级程度。智能化技术的逐步发展,为制导武器的智能升级带来了新的技术突破口。智能化技术将显著提升信息化系统的作战能力,若能有效突破精确制导系统的智能化技术应用瓶颈,将使现有的制导系统可以更好地适应复杂战场环境以及激烈对抗条件下的多类别目标精确打击需求。

基于深度学习的军事目标检测识别技术可以有效、自动、快速地识别战场目标,是作战双方利用智能技术理解战场态势的基础。智能技术在军事应用当中需要具备三大核心要素,包括深度学习算法设计、高性能智能计算平台以及大规模的数据训练集。首先建立深层次的神经网络模型;其次在规模庞大的数据集上进行预训练,并在战场数据集上进行模型的再次训练与微调;最终以实时处理模式在高性能计算平台上实现网络的实时推理计算,对多类型目标进行实时计算识别。

目前,基于卷积神经网络(Convolutional Neural Networks, CNN)的自主目标检测识别算法大致分为两类。一是基于区域建议的算法,包括区域建议卷积神经网络[5](Region-CNN, R-CNN)、Fast R-CNN[6]、Faster R-CNN[7]和区域建议全卷积神经网络[8](Region-based Fully Convolution Network, R-FCN),这类算法将目标识别与目标定位划分成 2 个步骤,分别完成,错误率低,但识别速度较慢;二是基于回归的算法,包括只看一次(You Only Look Once, YOLO)算法[9]、YOLO9000 算法[10]、单点多盒探测(Single Shot MultiBox Detector, SSD)算法[11]、去卷积单点探测(Deconvolu-

tional Single Shot Detector，DSSD）算法[12]、YOLOv2 算法和 YOLOv3 算法[13]等，该类算法直接产生目标类别概率和坐标，符合实时性要求，准确率也基本可以达到区域建议算法的准确率级别，可以在确保精度的同时，获得更高的时间效率。R-CNN 算法开创了深度学习自主目标识别的先河，Faster R-CNN 在 R-CNN 的基础上，直接提取候选区域特征图，并融入区域建议网络（Region Proposal Network，RPN），实现整个识别过程的网络统一，从而实现端到端之间的映射，大幅提升算法速度。在 YOLO 系列算法当中，YOLOv3 算法采用 Darknet53 的基础网络结构，在检测速度与精度两方面均获得了优于 SSD 系列算法的检测结果。目前，科研人员对于 YOLOv3 算法的落地应用拥有极高的研究热情。

由于智能算法复杂度较高，所以智能算法对于计算需求有着较高要求，这与嵌入式受限资源条件下的应用存在显著的矛盾，基于神经网络压缩的算法[14-15]应运而生。网络压缩算法将原本复杂度较高、参数冗余较多的网络，基于最优理论，在网络精度损失较小的情况下，压缩为复杂度较低、参数规模较小的网络结构，使其更加适应于资源受限条件下的硬件推理。目前，基于网络压缩的算法大致分为基于剪枝思想[16-17]的、基于张量分解思想的、基于权值共享思想的、基于权重量化思想[18-20]的、基于低比特或二值化思想[21-24]的压缩算法等。总而言之，网络压缩通过优化思想，减少网络参数，降低对硬件的资源需求，在对网络性能影响较小的情况下，实现智能算法在硬件端的实时推理部署。

本文首先基于复杂背景及小目标，分析了当前主流的自主检测识别网络，包括基于区域建议的方法和基于回归的方法，并对网络进行性能评估，使用 MAC 统计及参数需求量对其硬件需求进行定量评估，构成智能算法硬件嵌入式平台的基础输入要求。其次，提出了基于卷积神经网络的压缩算法，并对算法进行普适性分析。最后，基于嵌入式 GPU 平台，实现了基于 TensorRT 路线的神经网络加速，然后推理分析了经优化算法优化后的网络结构，并对网络精度损失情况和网络加速比情况进行了评价。

1 网络性能评估和硬件资源需求定量评估

对目前主流的深度学习自主识别算法进行参数及 MAC 计算量统计，如表 1 所示。可以看出，针对不同的神经网络，所需要的硬件资源各不相同。MAC 数目越多，代表硬件上所需的乘累加操作越多；权值合计越大，代表硬件上所需的存储空间越大。

表 1 深度学习识别算法参数及 MAC 计算量统计

Tab. 1 Deep learning recognition algorithms parameters and MAC calculated quantity statistics

算法	输入尺寸	MAC 数目合计	权值合计
Faster R-CNN（VGG16）	224×224	25.72G	87.53M
	448×448	31.59G	87.53M
R-FCN（ResNet101）	224×224	10.46G	46.05M
	448×448	41.85G	46.05M
YOLO	448×448	20.28G	271.7M
SSD（VGG16）	300×300	31.37G	26.28M
	512×512	90.21G	27.19M
DSSD	513×513	207.97G	254.98M
YOLOv3	416×416	65.43G	61.63M

从性能方面讲，在所有算法当中，YOLOv3 算法从速度和精度两方面均获得了较为惊艳的效果。Darknet53 借鉴残差结构，采用类似 ResNet 的跳线连接方式，性能相比 ResNet 系列更加优异。目前在各类落地应用当中，YOLO 系列算法更多采用 Tiny 网络，该网络层数较少，MAC 统计量约为 YOLOv3 算法的 1/13。Tiny 网络容易实现硬件应用，且仿真较为容易，但是网络精度相对较低，无法适应精度要求较高的多目标分类场合。于是，如何实现检测精度较高的 YOLOv3 算法的真正落地，是目前亟待解决的难题；此外，类似 YOLOv3 这类深层网络算法如何在嵌入式端落地，也是目前亟待解决的难题。

2 YOLOv3 网络结构与输出特征分析

YOLOv3 网络结构共计 107 层，网络最终输出三部分特征图，如图 1 所示，分别为 Conv_6、Conv_14 和 Conv_22 卷积节点，在此基础上，进行分类与位置回归。这 3 个卷积节点分别称之为小尺度 yolo 层、中尺度 yolo 层和大尺度 yolo 层。13×13×255 尺度用于检测较为大型的目标，26×26×255 用于检测较为中型的目标，52×52×255 用于检测较为小型的目标。三层特征输出层的详细输入、输出和卷积核参数如表 2 所示。

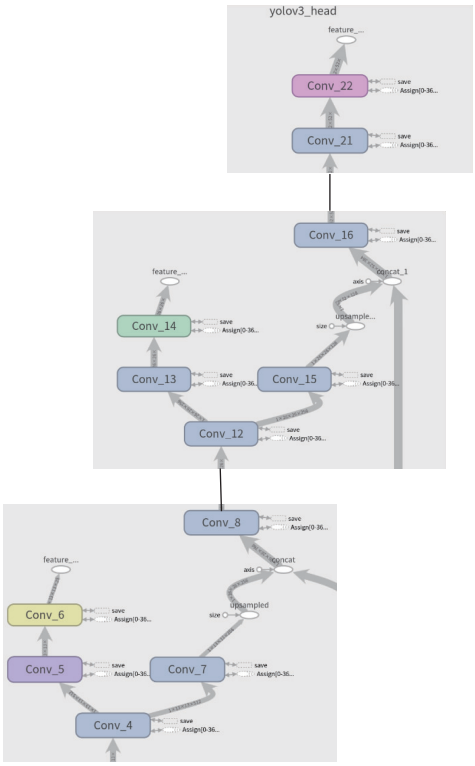

图 1　YOLOv3 网络结构的三部分特征输出

Fig. 1　Three-part feature outputs of YOLOv3 network structure

表 2　YOLOv3 网络输出特征图详细参数

Tab. 2　Detailed parameters of YOLOv3 network output characteristic diagram

节点	输入尺寸	输出尺寸	卷积核
Conv_6	1×13×13×1024	1×13×13×255	1×1×1024×255
Conv_14	1×26×26×512	1×26×26×255	1×1×512×255
Conv_22	1×52×52×256	1×52×52×255	1×1×256×255

3　神经网络裁剪算法

为神经网络设定合适的裁剪滤波器，从网络结构中剔除掉相对不重要的参数，将剩余网络结构进行微调或重新训练，可以在较短时间内有效对神经元或权重连接实现裁剪，网络裁剪过程如图 2 所示。

利用阈值方法对网络权重进行整体裁剪，是网络裁剪算法中最常用的步骤。假设阈值为 ω，保留每层中 filter 权重绝对值之和大于阈值 ω 的权重。阈值法裁剪方式如式(1)所示，其中 i 和 j 代表卷积核的维度

$$w'_{ij}=\begin{cases}w_{ij}, w_{ij}\geqslant\omega\\0, w_{ij}<\omega\end{cases}\tag{1}$$

式(1)很难从全局进行分析，并且不能将训练

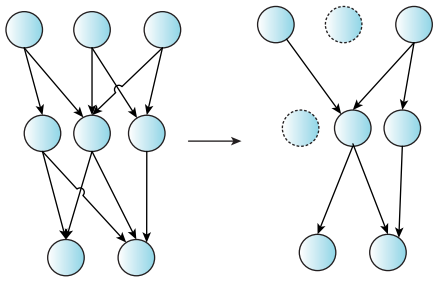

图 2　神经网络裁剪结构对比图

Fig. 2　Contrast diagram of pruning structure of neural network

融入裁剪当中，从而导致网络精度降低。文献[25]采用一种基于注意力模块的剪枝滤波器，称为 SE-Block，由全局池化层、全连接层和激活函数组成。将通过注意力模块的输出称为缩放因子，其变换过程描述如式(2)所示

$$F(X_{\langle n,W,H,C\rangle})=\mathrm{sig}(\mathrm{FC}_2(\mathrm{ReLU}(\mathrm{GAP}(X_{\langle n,W,H,C\rangle}))))\tag{2}$$

其中，$X_{\langle n,W,H,C\rangle}$ 代表输入，FC 代表全连接层，GAP 代表全局池化层，ReLU 和 sig 代表激活函数。

为了更好地使得裁剪过程自适应，本文提出将 SEBlock 与 BN 层缩放因子同时进行正则化训练裁剪的方法，算法策略如下所述：

1)通过 SEBlock 计算缩放因子，获得能够反映通道重要性的参数，结合通道在样本数据集下的平均值进行综合分析，更准确地反映通道重要性。

2)利用 L1 正则方法，在网络原本代价函数的基础上，将步骤 1)计算出的缩放因子，与 BN 层的缩放因子同时归入目标方程，进行稀疏化训练，如式(3)所示

$$L=\sum_{(x,y)}l(f(x,W),y)+\alpha\sum|I_c|+\beta\sum_{\gamma\in\Gamma}g(\gamma)\tag{3}$$

其中，第一项为网络原本代价函数，第二项为基于步骤 1)计算的缩放因子求和项，第三项为 BN 层缩放因子求和项，α 和 β 分别为正则项系数。$I_c=\frac{1}{N}\sum_{n=1}^{N}F(X_{\langle n,W,H,C\rangle})$，代表 SEBlock 缩放因子，$g(\gamma)=|\gamma|$，其中，$\gamma$ 代表 BN 层的缩放因子。

3)依照步骤 2)进行训练裁剪后，对网络进行微调，从而恢复裁剪后网络的检测精度。

网络裁剪的目标是在保持网络精度的前提下，保留重要权重，去掉不重要权重，其核心在于如何在裁剪的同时，更好地保持精度。文中所提出的自适应稀疏化训练方式可以在裁剪的同时，最大程度上保证网络精度。整个裁剪过程训练流程图如图 3

所示，首先利用式(3)对网络进行稀疏化训练，随后裁剪掉稀疏的网络连接，其次对裁剪后的网络进行微调，获得剪枝后的网络。此外，该训练过程还可重复进行，并不断迭代，以便获得最优结果。

图 3 裁剪过程训练流程图

Fig. 3 Pruning process training flow chart

本文提出的网络裁剪算法可以明显保持裁剪后的网络精度，针对本文裁剪算法已经经过测试的网络结构包括 VGG、ResNet、Darknet53 和 DenseNet 网络，该网络裁剪算法针对 CNN 具备普适性。

4 神经网络量化算法

最主流的权重量化方式包括 Fp16 量化和 Int8 量化，其中 Fp16 相比 Fp32 减少 50%的位宽，Int8 相比 Fp32 减少 75%的位宽。线性 Int8 量化将权重数据量化到(−127～127)的范围当中，这种映射称为不饱和映射，将导致精度损失较大。本文将采用饱和映射进行量化，这也是 TensorRT 技术采用的量化方式。饱和映射的过程是寻找阈值 $|T|$，将 $\pm|T|$ 映射到±127 范围当中，超过阈值之外的，直接映射到±127，饱和映射过程示意图如图 4 所示。

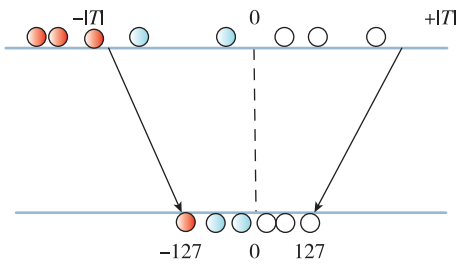

图 4 饱和映射过程示意图

Fig. 4 Schematic diagram of saturation mapping process

本文采用的保精度量化算法策略如下所述：

1)从验证集当中，选取子集当作校准集，用于校准 Int8 量化带来的精度损失；

2)在选取的校准集上进行 Fp32 推理，对于网络的所有层，分别收集相关的激活值，列出直方图；

3)针对不同阈值实施遍历操作，选取可以使得 KL 散度取得最小值的阈值，最终获得一系列的阈值，并且所有层均返回一个阈值，称之为校准表(Calibration Table)，最终利用校准表实现神经网络 Int8 的保精度量化过程。

本文采用的网络量化算法可以明显保持量化后的网络精度，针对本文所采用的量化算法，已经经过测试的网络结构包括 VGG、ResNet、Darknet53、AlexNet 和 GoogleNet 网络，该网络量化算法针对 CNN 具备普适性。

5 TensorRT 神经网络优化技术

TensorRT 技术属于英伟达的不开源神经网络加速技术，为神经网络部署提供基于 GPU 平台的加速解决方案。目前，TensorRT 技术最擅长 CNN 优化，TensorRT 技术程序部署流程如图 5 所示。

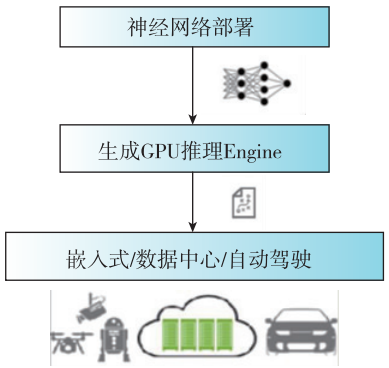

图 5 TensorRT 程序部署流程

Fig. 5 Program deployment process based on TensorRT

TensorRT 通过对网络进行合并与量化，形成更为紧凑、硬件资源需求更小的网络结构，能够确保在减小资源使用率的同时，使得网络结构性能损失程度较小。TensorRT 技术首先通过优化技术生成如图 5 所示的中间层 engine，随后利用该优化后的 engine 对网络结构进行部署，实现各类受限资源条件下的神经网络实时应用。

对于如图 6 所示的原始网络，TensorRT 技术可将其垂直方向优化为如图 7 所示的优化后网络结构，从而有效实现网络推理加速，水平方向的优化与垂直方向类似。此外，结合网络裁剪技术，可在推理过程当中获得更高的加速比。

6 仿真验证

为了验证本文提出和采用的网络压缩算法在嵌入式端的加速能力，选取 Nvidia Jetson Xavier 作为验证平台，并与 TensorRT 优化进行比对。Xavier 是英伟达的异构嵌入式 GPU 平台，CPU 具

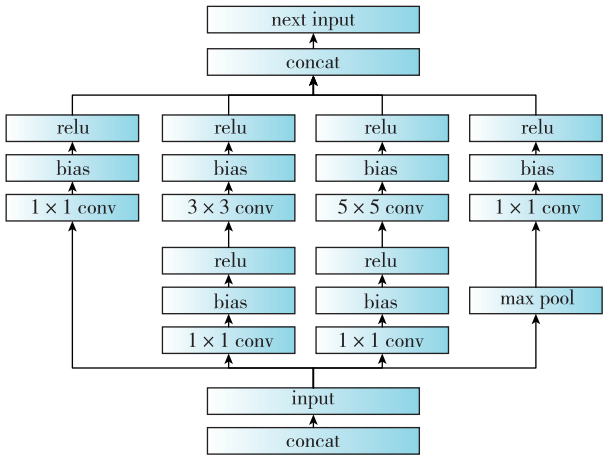

图 6　原始网络结构

Fig. 6　Original network structure

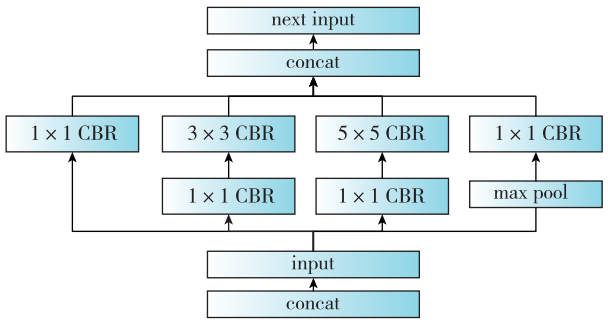

图 7　垂直方向优化后的网络结构

Fig. 7　Vertical optimized network structure

备 8 核 ARM64 架构，GPU 具备 512 颗 CUDA 核心。在公有数据集 VOC2007 与 VOC2012 上进行数据训练，利用获得的权重计算初始精度 mAP 值。随后利用裁剪与量化算法进行优化，并利用经算法优化后的权重计算新的 mAP 值。经过 20 次裁剪与量化仿真测试，选取最优仿真结果，同时，对未开源的 TensorRT 技术进行技术应用，仿真验证结果如表 3 所示。

表 3　YOLOv3 算法嵌入式端仿真验证结果

Tab. 3　Simulation and verification results of YOLOv3 on embedded GPU platform

测试类别	输入尺寸	精度 mAP	帧频	加速方式
darknet 框架	416×416	0.7503	8 帧/s	未加速
TensorRT 加速	416×416	未开源，无法测试	25 帧/s	Int8
裁剪	416×416	0.7401(−1.36%)	12 帧/s	裁剪
裁剪＋量化	416×416	0.7152(−4.68%)	26 帧/s	裁剪＋Int8

YOLOv3 算法的部分网络裁剪结果如表 4 所示。由表 4 可以看出，裁剪算法针对不同的卷积层会进行相应的裁剪，有些层通道数目变小，有些层通道数目不变。网络裁剪将直接改变网络结构，裁剪结果与训练数据集密切相关。

表 4　YOLOv3 算法部分网络裁剪结果

Tab. 4　Partial network pruning results of YOLOv3

层数	输入	卷积核	通道	裁剪后通道
1	416×416×32	3×3×32×64	32	25
2	208× 208 × 64	1×1×64×32	64	64
3	208× 208 × 32	3×3×32×64	32	21
5	208× 208 × 64	3×3×64×128	64	64
6	104× 104 × 128	1×1×128×64	128	128
7	104× 104 × 64	3×3×64×128	64	10
9	104× 104 × 128	1×1×128×64	128	128

YOLOv3 算法裁剪前后的精度与帧频对比结果如表 5 所示，表中结果均是分别在相应条件下运行 10 次程序后选取的最优结果。可以看出，针对不同的裁剪力度，精度下降情况和帧频变化均不同，裁剪力度越大，精度下降越快，帧频越高。

表 5　YOLOv3 算法裁剪前后精度与帧频对比

Tab. 5　Accuracy and frame frequency comparison of YOLOv3 before and after pruning

裁剪方式	精度 mAP	帧频
未裁剪	0.7503	8 帧/s
权重减少 10%	0.7475(−0.37%)	9 帧/s
权重减少 15%	0.7452(−0.68%)	10 帧/s
权重减少 20%	0.7401(−1.36%)	12 帧/s

从表 3～表 5 的仿真验证结果可以看出，裁剪与量化的方式可以在网络精度损失较少的情况下，使得嵌入式平台上的网络推理获得理想的加速比。原版 darknet 在异构嵌入式平台的帧频为 8 帧/s，经 TensorRT 技术优化后可获得 3 倍的速度提升。经本文提出的网络压缩算法，可以在精度损失小于 5%的前提下，获得 3 倍以上的速度提升。

相比不开源的 TensorRT 技术，本文算法思想可以实现自主可控的神经网络压缩及嵌入式应用。此外，基于本文思想，结合复杂算法实现裁剪与量化，将使得目标检测识别网络精度下降幅度更小。

7　结论

本文提出了针对卷积神经网络的压缩算法，并

进行了相应的嵌入式平台应用。相比不开源的针对 GPU 平台的 TensorRT 优化技术，本文算法思想可以合理进行各类硬件平台的技术复用。针对神经网络的定量硬件资源评估，以及针对 GPU 嵌入式平台所进行的裁剪和量化实验分析表明：

1)各类自主目标识别神经网络算法的硬件资源需求量可通过计算获得，针对目标算法，可以利用资源计算分析来合理设计硬件。

2)基于英伟达目前的 TensorRT 技术，利用 8bit 量化技术，在嵌入式 GPU 平台可以实现神经网络 3 倍的推理速度提升。由于此项技术为不开源技术，所以精度损失程度未知。从公开资料来看，网络精度损失较小。

3)基于本文所提出和采用的网络裁剪及量化优化算法，在网络精度损失小于 5%的前提下，获得了 3 倍以上的推理速度提升。本文算法与针对 GPU 平台的不开源 TensorRT 技术相比，为针对不同平台的神经网络优化技术应用提供了新的技术思路。

经验证，本文的优化算法思想可直接应用于 FPGA 平台。本文的下一步研究方向是基于 FPGA 平台的硬件优化[26-27]。在具备国产自主性、低功耗的 FPGA 平台，利用网络优化技术实现神经网络的实时应用部署，为制导武器末制导人工智能机器视觉技术的军事应用提供进一步的技术解决方案。

参考文献

[1] Sayrol E, Gasull A, Fonollosa J R. Motion estimation using higher order statistics[J]. IEEE Transaction on Image Processing, 2006, 5(6): 1077-1084.

[2] Manchanda S, Sharma S. Identifying moving objects in a video using modified background subtraction and optical flow method[C]// Proceedings of 3rd International Conference on Computing for Sustainable Global Development. IEEE, 2016.

[3] Attamimi M, Nakamura T, Nagai T. Hierarchical multilevel object recognition using Markov model [C]// Proceedings of the 21st International Conference on Pattern Recognition, 2012.

[4] Elakkiya S, Audithan S. Feature based object recognition using discrete wavelet transform[C]// Proceedings of 2nd International Conference on Current Trends in Engineering and Technology (ICCTET). IEEE, 2014.

[5] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014: 580-587.

[6] Girshick R. Fast R-CNN[C]// Proceedings of IEEE International Conference on Computer Vision, 2015: 1440-1448.

[7] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[8] Dai J F, Li Y, He K M, et al. R-FCN: object detection via region-based fully convolutional networks [C]// Proceedings of 30th Conference on Neural Information Processing Systems (NIPS), 2016.

[9] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.

[10] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6517-6525.

[11] Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector [C]// Proceedings of European Conference on Computer Vision (ECCV), 2016: 21-37.

[12] Fu C Y, Liu W, Ranga A, et al. DSSD: deconvolutional single shot detector[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[13] Redmon J, Farhadi A. YOLOv3: an incremental improvement[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[14] Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding[C]// Proceedings of International Conference on Learning Representations, 2016.

[15] Chen W L, Wilson J T, Tyree S, et al. Compressing neural networks with the hashing trick[C]// Proceedings of International Conference on Machine Learning, 2015: 2275-2284.

[16] Yu R C, Li A, Chen C F, et al. NISP: pruning networks using neuron importance score propagation [C]// Proceedings of IEEE Transactions on Com-

puter Vision and Pattern Recognition, 2018: 9194-9203.

[17] Liu Z, Li J G, Shen Z Q, et al. Learning efficient convolutional networks through network slimming [C]// Proceedings of International Conference on Computer Vision, 2017: 2755-2763.

[18] Gupta S, Agrawal A, Gopalakrishnan K, et al. Deep learning with limited numerical precision[C]// Proceedings of International Conference on Machine Learning, 2015: 1737-1746.

[19] Wu J X, Leng C, Wang Y H, et al. Quantized convolutional neural networks for mobile devices[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016: 4820-4828.

[20] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2704-2713.

[21] Lin X F, Zhao C, Pan W. Towards accurate binary convolutional neural network [C]// Proceedings of 31st Annual Conference on Neural Information Processing Systems (NIPS), 2017.

[22] Cai Z W, He X D, Sun J, et al. Deep learning with low precision by half-wave gaussian quantization [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017: 5406-5414.

[23] Courbariaux M, Bengio Y, David J P. Binary Connect: training deep neural networks with binary weights during propagations [C]// Proceedings of Annual Conference on Neural Information Processing Systems(NIPS), 2015.

[24] Leng C, Dou Z, Li H, et al. Extremely low bit neural network: squeeze the last bit out with ADMM[C]// Proceedings of 32nd AAAI Conference on Artificial Intelligence, 2018: 3466-3473.

[25] 汪枭杰. 卷积神经网络压缩技术的研究与实现[D]. 北京: 北京邮电大学, 2019.
Wang Xiaojie. Research and implementation of convolutional neural network compression technique[D]. Beijing: Beijing University of Posts and Telecommunications, 2019(in Chinese).

[26] Han S, Kang J L, Mao H Z, et al. ESE: efficient speech recognition engine with sparse LSTM on FPGA[C]// Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2017: 75-84.

[27] Zhang C, Li P, Sun G Y, et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks[C]// Proceedings of 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2015: 161-170.